การประยุกต์ใช้ เทอร์โบปาสคาล 4.0-5.0 กับ CAI (ตอนจบ)

อาจหาญ สัตยารักษ์

การเรียนรู้กราฟิกบนปาสคาล 4.0-5.0 ที่ได้ศึกษามาทั้ง 4 ตอนนั้น นับว่าสามารถนำไปเขียนกราฟิกต่าง ๆ ได้ค่อนข้างสมบูรณ์แล้ว ดังนั้นจึงจะไม่อธิบายรูปแบบของคำสั่งอื่น ๆ เพิ่มเติมอีก หากมีคำสั่งใดที่ผู้เขียนไม่ได้พูดถึงก็ให้ศึกษาได้จากหนังสือคู่มือคงไม่ยากนัก

สำหรับตอนนี้เราจะมาพูดถึงการเก็บภาพที่เราได้วาดหรือสร้างแล้วลงในแผ่นดิสค์กัน

การเก็บรูปกราฟิก

เมื่อเราได้สร้างรูปภาพบนจอภาพไว้แล้ว เราสามารถเก็บภาพนั้นลงในแผ่นดิสค์ได้ในรูปแบบของการเก็บเท็กซ์ไฟล์ทั่ว ๆ ไป แต่สิ่งสำคัญอยู่ที่เราต้องเก็บภาพลงในตัวแปรมิติตัวใดตัวหนึ่งก่อน(สมมติผู้เขียนเก็บในตัวแปรชื่อ Picbuff และเป็นตัวแปรชนิดไบต์) สิ่งแรกที่เราต้องคิดคือขนาดของมิติว่าจะมีค่าเป็นเท่าใดจึงจะเก็บภาพที่เราสร้างไว้ได้พอดี

การคำนวณหาขนาดของมิตินี้ สำหรับจอโมโนโครมเราสามารถหาได้จากสูตร

array = 4 + INT (((x2-x1)+8)/8)*(y2-y1+1)

เมื่อ x1,y1 คือมุมบนซ้ายของภาพ

x2,y2 คือมุมล่างขวาของภาพ

เช่น ต้องการเก็บภาพหมดทั้งหน้าจอภาพ จะได้ 0,0 คือ x1,y1 และ 719,347 คือ x2,y2 ดังนั้น จึงหาค่าได้จาก

array = 4 + INT (((719-0)+8)/8)*(347-0+1)
= 4 +INT(727/8)*348
= 4 + 90*348
= 31324 ไบต์

หรือต้องการเก็บภาพที่สร้างไว้ในกรอบ 30,20,100,60 ก็จะได้ขนาดของมิติดังนี้

array = 4 + INT (((100-30)+8)/8)*(60-20+1)
= 373 ไบต์

นำค่าที่คำนวณได้นี้ไปกำหนดขนาดของตัวแปร Picbuff ดังนั้นโดยวิธีนี้เราจึงสามารถเก็บภาพขนาดเท่าใดก็ได้

สร้างภาพโดยการกดแป้นพิมพ์

เราจะอาศัยแนวคิดข้างต้นมาเก็บภาพที่เราวาดไว้เพื่อนำไปใช้งานในภายหลัง ในการวาดภาพนั้น เราอาจจะเขียนชุดคำสั่งขึ้นมาช่วย หรือจะเขียนแบบแนวคิดเดิมที่ผ่านมาก็ได้ ในที่นี้เราจะลองมาใช้การกดแป้นพิมพ์ประมาณ 17 แป้น เพื่อให้เกิดภาพบนจอภาพแล้วนำภาพ(ที่มีขนาดเท่ากัน)มาเก็บลงในเท็กซ์ไฟล์โดยมีโปรแกรมดังรูปที่ 1

จากโปรแกรมที่ 1 (ในรูปที่ 1) จะเป็นการวาดภาพในกรอบขนาด 360,50,435,100 ดูได้จากคำสั่ง getimage(360, 50,435,100,Picbuff) ในชุดคำสั่ง SavePic ซึ่งจากกรอบดังกล่าวจึงคำนวณขนาดของมิติได้เท่ากับ 514 ไบต์ เราจึงกำหนดขนาดของภาพเป็น array[1...514] of byte;

ตัวแปรที่กำหนดไว้ในโปรแกรมที่ควรทราบมีดังนี้

Picbuff รูปภาพ(ขนาด 514 ไบต์)ที่จะเก็บลงในไฟล์ (ชื่อ Graphic.pic)

RecordNo ลำดับของภาพที่เก็บไว้ในไฟล์ (โดยภาพที่ 1 เก็บไว้เป็นลำดับที่ 0)

Piccur ภาพของเคอร์เซอร์เพื่อให้ดูออกว่ากำลังวาดภาพถึงตอนไหน

XDraw,YDraw จุดที่เคอร์เซอร์กะพริบอยู่

XPix,YPix จุดของรูปภาพที่วาดได้จริง

แป้นที่ใช้เมื่ออยู่ในชุดคำสั่ง DrawPix มีดังนี้

1. แป้นที่ใช้เลื่อนเคอร์เซอร์ มี 4 แป้น คือ แป้นลูกศรซ้ายขวา และลูกศรขึ้นลง (ไม่มีข้อความบอกบนจอภาพ)

2. แป้น D ใช้เมื่อต้องการลงจุด ณ ตำแหน่งที่เคอร์เซอร์กะพริบอยู่ โดยการลงจุดนี้จะทำพร้อมกันทั้งสองแห่ง คือที่กรอบใหญ่ด้านซ้ายที่ใช้เป็นแนวในการวาดรูป กับที่ด้านขวามือซึ่งส่วนนี้จะให้ภาพขนาดเท่าของจริงเมื่อนำไปใช้งาน

3. แป้น E ใช้เมื่อต้องการลบจุด ณ เคอร์เซอร์กะพริบ ซึ่งจะลบทั้งสองรูปแบบเดียวกับการวาด

4. แป้น P ใช้เมื่อต้องการระบายสีลงในส่วนที่เป็นรูปปิด (ถ้ารูปไม่ปิดจะมีผลทำให้การระบายเลอะเต็มจอภาพได้ จึงต้องดูให้ดีทุกครั้งก่อนจะกดแป้น P)

5. แป้นตัวเลข 1 ถึง 9 ใช้เลือกรูปแบบของการระบายสี 9 แบบ ใช้คู่กับแป้น P เมื่อกดแป้นกลุ่มนี้ จะมีภาพเป็นกรอบสี่เหลี่ยมเล็ก ๆ แสดงรูปแบบนั้น ๆ ให้เห็นด้วย(ดูรูปที่ 2 ประกอบ)

6. แป้น Q ใช้เมื่อสิ้นสุดการวาดภาพ ซึ่งจะเป็นการจบชุดคำสั่ง DrawPix นี้ เพื่อไปสู่การเก็บภาพลงแผ่นดิสค์ต่อไป

ชุดคำสั่ง CreatFile ใช้เพื่อทำการสร้างไฟล์ในชื่อที่กำหนด และทำการเปิดไฟล์ที่สร้างนี้ไว้เพื่อเตรียมเก็บภาพ หากเป็นการเขียนภาพต่อจากที่เคยสร้างไว้ก็ให้ลบบรรทัดที่มี Rewrite ออกไป (พร้อมกับกำหนดค่า RecordNo ให้ต่อจากที่เคยเขียนไปแล้วด้วย)

ชุดคำสั่ง SavePic ใช้เมื่อต้องการเก็บภาพลงในลำดับที่ต้องการ(ค่า RecordNo) พร้อมกับเพิ่มค่า RecordNo เพิ่มขึ้นอีก 1 ความจริงอาจเขียนให้ตรวจสอบลำดับสุดท้ายของข้อมูลโดยไม่กำหนด RecordNo ก็ได้ แต่ในโปรแกรมนี้ผู้เขียนกำหนดไว้เพื่อนำไปแสดงบนจอภาพว่าขณะนั้นกำลังวาดภาพลำดับที่เท่าใด (Pic No.)

เมื่อเก็บภาพที่วาดแล้วโปรแกรมจะถามว่าจะวาดต่อหรือเลิก ภาพที่วาดได้จากโปรแกรมในรูปที่ 1 ดูได้ในรูปที่ 2

การเรียกภาพจากไฟล์มาแสดงผลบนจอภาพ

เมื่อเก็บภาพไว้หลายภาพแล้วก็มาถึงการนำภาพเหล่านั้นมาแสดงบนจอภาพ เราทำได้โดยการอ่านภาพที่เก็บไว้แล้วนั้นเข้ามาเก็บไว้ในตัวแปรแบบเดิม(Picbuff) แต่เพื่อสะดวกในการนำภาพมาแสดงผล เราก็กำหนดเงื่อนไขหรือตัวแปรเพิ่ม เช่น

กำหนดค่า x เพื่อให้เป็นจุดเริ่มต้นของภาพตามแนวตั้ง
y เพื่อเป็นจุดเริ่มต้นของภาพตามแนวนอน
PicNo ลำดับของภาพที่ต้องการเรียกมาปรากฏบนจอภาพ
Mode รูปแบบการนำภาพมาแสดง มีค่าระหว่าง 0 ถึง 4

ด้วยตัวแปร x,y,PicNo เราก็สามารถนำภาพที่เก็บไว้นั้นไปแสดงในตำแหน่งที่ต้องการได้ด้วยคำสั่ง Putimage ดังนี้

```
          {----- PROGRAM 5.1 DrawPic ------}
    {$I- INITGR.PAS}
Type Pic=array [1..514] of byte;
     PicRecord=RECORD
           Picbuff:pic;
           end;
     FileType = file of PicRecord;
var  k,x,y,XPix,YPix,XDraw,YDraw:integer;
     PicFile:filetype; Ch:char;
     RecordNo,Endcode:byte;
     Piccur:pointer; Size:word;

PROCEDURE DrawPix;
var Code,Pc:byte; st:string[3];
begin
  setfillstyle(1,0); bar(10,10,450,300);
  str(RecordNo+1,st); outtextxy(350,10,'Pic no.'+st);
  outtextxy(350,120,'D :Draw');
  outtextxy(350,140,'E :Erase');
  outtextxy(350,160,'P :Paint');
  outtextxy(334,180,'1-9 :Fillstyle');
  outtextxy(350,200,'Q :Quit');
  setfillstyle(1,1);  bar(465,175,495,195);
  Rectangle(9,9,311,211);  size:=imagesize(0,0,4,4);
  getmem(Piccur,size);  Xpix:=10; Ypix:=10;
  code:=0; getimage(Xpix,ypix,Xpix+3,Ypix+3,Piccur^);
  XDraw:=360; YDraw:=50; Pc:=1;
 Repeat
   putimage(Xpix,Ypix,Piccur^,4); delay(70);
   putimage(Xpix,Ypix,Piccur^,0); delay(70);
  if keypressed then begin
     ch:=readkey;
      if Upcase(ch)='D' then                          begin
        setfillstyle(1,1); bar(Xpix,Ypix,XPix+3,YPix+3);
        putPixel(Xdraw,YDraw,1);
        getimage(Xpix,ypix,Xpix+3,Ypix+3,Piccur^);  end;
      if Upcase(ch)='E' then                           begin
        setfillstyle(1,0); bar(Xpix,Ypix,XPix+3,YPix+3);
        getimage(Xpix,ypix,Xpix+3,Ypix+3,Piccur^);
        putPixel(Xdraw,YDraw,0); setfillstyle(1,1);  end;
      if Upcase(ch)='P' then                         begin
        setfillstyle(Pc,1);Floodfill(Xpix,Ypix,blue);
        Floodfill(Xdraw,YDraw,blue);
        getimage(Xpix,ypix,Xpix+3,Ypix+3,Piccur^); end;
      if (ch>'0') and (ch<='9') then                          begin
        Pc:=ord(ch)-48; setfillstyle(Pc,1);
        bar(465,175,495,195); Rectangle(465,175,495,195); end;
      if Upcase(ch)='Q' then code:=27;
```

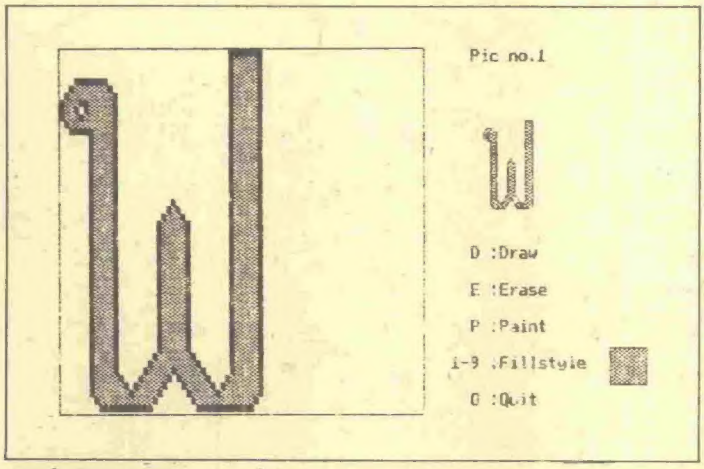
รูปที่ 1
โปรแกรมสร้างภาพด้วยการกดแป้นพิมพ์

```
        if ch=#0 then       begin    ch:=readkey;
             case ch of
              #77:if Xpix<306 then                    begin
                     Xpix:=Xpix+4; Xdraw:=Xdraw+1; end;
              #75:if Xpix>10 then                     begin
                     Xpix:=Xpix-4; Xdraw:=Xdraw-1; end;
              #80:if Ypix<206 then                    begin
                     Ypix:=Ypix+4; Ydraw:=Ydraw+1; end;
              #72:if Ypix>10 then                     begin
                     Ypix:=Ypix-4; Ydraw:=Ydraw-1; end;
             end; {case}
              getimage(Xpix,ypix,Xpix+3,Ypix+3,Piccur^);
                        end; {ch=#0}
                    end;  {keypressed }
 until code=27;
end;

PROCEDURE CreatFile(Filename:string);
 begin
     Assign(PicFile,FileName);
     Rewrite(PicFile);
     Reset(PicFile);
 end;  { CreatFile }

PROCEDURE SavePic;
var   PicRec:PicRecord;
 BEGIN
   Seek(PicFile,RecordNo);
    With PicRec Do  getimage(360,50,435,100,Picbuff);
   Write(PicFile,PicRec);
   RecordNo:=RecordNo+1;
 end; { SavePic }

 {====================== Main ======================}

  begin
   opengraph;  endcode:=0; RecordNo:=0;
   CreatFile('Graphic.pic');
   Repeat
    DrawPix;    SavePic;
    outtextxy(50,250,'Do you want to Draw other Pic (Y/N)');
      Repeat
        Ch:=readkey;  if upcase(ch)='N' then Endcode:=27;
      Until (upcase(ch)='Y') or (upcase(ch)='N');
   Until Endcode=27;
   close(PicFile);   closegraph;
  end.
```

Putimage(x,y,Picbuff,Mode);

ผู้เขียนได้ลองใช้โปรแกรมในรูปที่ 1 มาวาดภาพเก็บไว้ในไฟล์ชื่อ Graphic.pic รวม 5 ภาพคือ

ภาพอักขระ ก เก็บไว้ลำดับที่ 0



รูปที่ 2 ตัวอย่างภาพที่สร้างได้จากโปรแกรมในรูปที่ 1

ภาพอักขระ ร เก็บไว้ลำดับที่ 1

ภาพอักขระ า เก็บไว้ลำดับที่ 2

ภาพอักขระ ฟ เก็บไว้ลำดับที่ 3

ภาพอักขระ ิ เก็บไว้ลำดับที่ 4

เมื่อต้องการนำภาพทั้ง 5 ภาพนี้ มาแสดงประกอบกันให้เกิดข้อความว่า **กราฟิก** บนจอภาพ ก็ทำการเขียนชุดคำสั่ง ShowPic ดังโปรแกรมในรูปที่ 3 โดยการส่งค่าตัวแปร x,y และ PicNo ไปให้เรียกภาพมาแสดง ณ จุดที่กำหนด รายละเอียดของโปรแกรมคิดว่าคงทำความเข้าใจได้ไม่ยาก จึงขอไม่อธิบาย แต่จะสังเกตได้ว่าทุกครั้งภาพจะถูกเรียกมาที่ตัวแปร Picbuff และให้แสดงบนจอภาพในทันที ซึ่งเมื่อรันก็จะได้ภาพปรากฏบนจอภาพดังรูปที่ 4

ประยุกต์สร้างภาษาไทยกราฟิก

จากแนวทางที่ผ่านมาถ้าเราสร้างภาพอักขระภาษาไทยตั้งแต่ ก ถึง ฮ สระ วรรณยุกต์ ไว้ให้ครบถ้วนแล้ว ก็น่าจะ

สามารถนำภาพเหล่านี้มาแสดงบนจอภาพให้เป็นข้อความภาษาไทยได้ โดยต้องกำหนดขอบเขตของภาพของอักขระที่ต้องการให้เหมาะสมด้วย

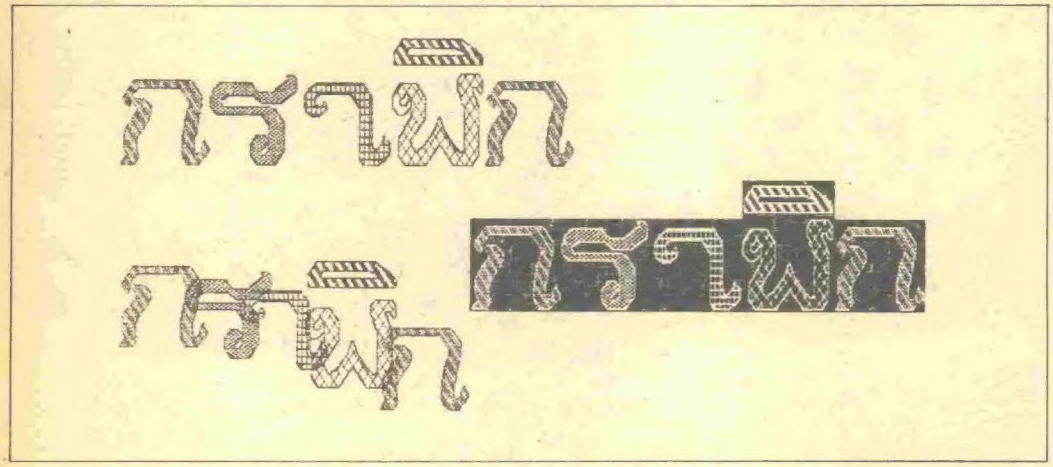
เมื่อมองเห็นทางในการสร้างภาษาไทยกราฟิกแบบนี้ เราก็ดัดแปลงโปรแกรมในรูปที่ 1 ให้มีขอบเขตของภาพตรงกับที่ต้องการได้

สมมติผู้เขียนเลือกใช้ตารางขนาด 14×22 จุด มาเป็นขนาดของอักขระแต่ละตัว ซึ่งทำให้ได้ขนาดมิติเป็น 48 ไบต์ ก็เริ่มเขียนภาพของอักขระเก็บไว้ในไฟล์ชื่อ THAI.CHR เรียงลำดับตามรหัสภาษาไทยเกษตรรวมทั้งสิ้น 69 ภาพ(ไฟล์ thai.chr นี้มีอยู่ในแผ่นซอฟต์เมล์) ก็ได้ภาษาไทยครบทุกอักขระ

การทำให้ภาษาไทยกราฟิกที่เราสร้างไว้มาแสดงผลบนจอภาพก็ทำได้ โดยกำหนดให้ภาพของอักขระเหล่านั้น มาอยู่ในตัวแปรมิติเรียงลำดับตามอักขระ ก ถึง ์ สมมติใช้ชื่อว่า THAI โดยจะได้ Thai[0] เป็นช่องว่าง Thai[1] เป็น ก Thai[68] เป็น ์ ซึ่งตัวแปรเหล่านี้ก็เป็นตัวแปรชนิดไบต์แบบเดียวกับ Picbuff (ตัวแปร Picbuff เป็นตัวแปรที่กำหนดไว้ใน RECORD จึงไม่สามารถนำมาใช้งานข้างนอกได้ ต้องกำหนด

```
                {---- PROGRAM 5.2 ShowPic-----}
     {$I INITGR.PAS}
Type Pic=array [1..514] of byte;
     PicRecord=RECORD
           Picbuff:pic;
           end;
     FileType = file of PicRecord;
var PicFile:Filetype;

PROCEDURE OpenFile(Filename:string);
 begin
     Assign(PicFile,FileName);
     Reset(PicFile);
 end;   {OpenFile }

PROCEDURE ShowPic(x,y:integer; PicNo,Mode:byte);
var   PicRec:PicRecord;
 BEGIN
   Seek(PicFile,PicNo);
   Read(PicFile,PicRec);
   With PicRec Do   Putimage(x,y,Picbuff,Mode);
 end; { ShowPic }

 {===================== Main ======================}
  begin
   opengraph;
   OpenFile('Graphic.pic');
     ShowPic(40,52,0,1);        ShowPic(115,52,1,1);
     ShowPic(190,52,2,1);       ShowPic(265,52,3,1);
     ShowPic(265,0,4,1);        ShowPic(340,52,0,1);
     ShowPic(40,152,0,2);       ShowPic(95,158,1,2);
     ShowPic(140,164,2,2);      ShowPic(195,174,3,2);
     ShowPic(195,120,4,2);      ShowPic(255,186,0,2);
     ShowPic(330,130,0,4);      ShowPic(405,130,1,4);
     ShowPic(480,130,2,4);      ShowPic(555,130,3,4);
     ShowPic(555,79,4,4);       ShowPic(630,130,0,4);
     setfillstyle(1,0); bar(550,79,630,108); setfillstyle(1,1);
   close(PicFile);
   waittogo;    closegraph;
  end.
```

รูปที่ 3
โปรแกรมเรียกภาพที่สร้างไว้แล้วมาแสดงผลงานบนจอภาพ



รูปที่ 4
ภาพที่ได้จากโปรแกรมในรูปที่ 3

```
                {---- PROGRAM 5.3 THAIFONT-----}
     {$I INITGR.PAS}
 Type Pic=array [1..48] of byte;
     PicRecord=RECORD
           Picbuff:pic;
           end;
     FileType = file of PicRecord;
 var PicFile:Filetype; k:byte;
     Thai:array[0..68] of Pic;

 PROCEDURE SaveThai;
 var   PicRec:PicRecord;
  BEGIN  k:=0;
     Assign(PicFile,'THAI.CHR');
     Reset(PicFile);
   Repeat
     Seek(PicFile,k);       Read(PicFile,PicRec);
     With PicRec Do Thai[k]:=PicBuff;     k:=k+1;
   until k=69;
     Close(PicFile);
 end; { SaveThai }

 PROCEDURE Outthai(x,y:integer; content:string);
 var asc:byte;
     yy:integer;
 begin
    Textshow(2,7,'h');
    for k:=1 to length(Content) do
     begin
      yy:=0;
      if content[k]<#127 then outtextxy(x,y,content[k]);
      if content[k]>#159 then                begin
          asc:=ord(content[k])-160;
          if content[k]>#214 then begin x:=x-13; yy:=17; end;
          if content[k]>#216 then  yy:=-16;
          if content[k]>#223 then
            if content[k-1]<#217 then yy:=-9;
          Putimage(x,y+yy,Thai[asc],1);    end;
       x:=x+13;
     end;
 end; { Outthai }

{==================== Main ======================}
 begin
  opengraph;
  SaveThai;
  Outthai(100,20,'ภาษาไทย + ภาษาอังกฤษ (Thai & English)');
  Outthai(70,50,'โดย อาจหาญ สัตยารักษ์ โรงเรียนลำปางกัลยาณี 52000');
  bar(0,80,719,81);
  Outthai(30,90,'สร้างโดยอาศัยการเขียนกราฟิก ที่พัฒนาจากโปรแกรมที่ 1');
  outthai(10,120,'ในกรอบกราฟิกขนาด 14 x 22 จุด ทำให้หนึ่งตัวอักขระใช้หน่วยความจำ');
  outthai(10,150,'48 ไบต์ โดยสร้างเก็บไว้ในไฟล์ชื่อ THAI.CHR');
  outthai(30,180,'โดยการอ่านรูปภาพจากไฟล์ THAI.CHR  มาเก็บไว้ในตัวแปร THAI');
  outthai(10,210,'ตั้งแต่ตัวแปรที่ 0-68 ( THAI[0]-THAI[68] )');
  outthai(30,240,'สามารถนำมาแสดงผลภาษาไทยบนจอภาพได้ และเมื่อใช้คู่กับภาษาอังกฤษ');
  outthai(10,270,'(ไฟล์LITT.CHR)ก็ทำให้แสดงผลภาษาไทยและภาษาอังกฤษได้อย่างสมบูรณ์');
  outthai(30,320,'โดยแนวคิดเดียวกันนี้ เราสามารถสร้างภาษาไทยในรูปแบบอื่นๆได้');
  repeat until keypressed;
  closegraph;
 end.
```

รูปที่ 5
โปรแกรมประยุกต์การสร้างภาพเพื่อทำภาษาไทยกราฟิก

ใหม่ให้เป็นตัวแปรมิติ Thai ก่อน)

โปรแกรมในรูปที่ 5 แบ่งได้เป็น 2 ชุดคำสั่ง คือ

-ชุดคำสั่ง SaveThai ทำหน้าที่ในการเรียกภาพที่อยู่ในไฟล์ Thai.chr มาเก็บในตัวแปร Thai[0]-Thai[68]

-ชุดคำสั่ง OutThai ทำหน้าที่นำข้อความที่กำหนด(ตัวแปรชื่อ Content) มาแสดง ณ จุดเริ่มต้น(ตัวแปร x,y) ข้อความที่ป้อนนี้ใช้รหัสเกษตร โดยการตรวจจากค่าแอสกีของอักขระ(ตัวแปรที่ใช้คือ asc โดยลบออกด้วย 160 เพื่อให้ได้ลำดับตรงกับภาพที่เก็บไว้) ก็สามารถนำภาพไปแสดง ณ ตำแหน่งที่ถูกต้องของภาษาไทยได้

ตำแหน่งที่ต้องตรวจสอบก็คือ content[k]>#214 คือ สระ(เฉพาะสระบนและสระล่าง) และวรรณยุกต์ทั้งหมด ต้องกำหนดค่า x ขยับไปทางซ้าย (คือ x:=x−13;) พร้อมกับกำหนดค่า yy ให้เหมาะสมด้วย

content[k]>#216 คือ สระบนและวรรณยุกต์ ต้องกำหนดค่า yy ใหม่

ภาษาไทย + ภาษาอังกฤษ (Thai & English)
โดย อาจหาญ สัตยารักษ์ โรงเรียนลำปางกัลยาณี 52000

สร้าง โดยอาศัยการเขียนกราฟิก ที่พัฒนาจากโปรแกรมที่ 1 ในกรอบกราฟิกขนาด 14 x 22 จุด ทำให้หนึ่งตัวอักขระ ใช้หน่วยความจำ 48 ไบต์ โดยสร้าง เก็บไว้ในไฟล์ชื่อ THAI.CHR

โดยการอ่านรูปภาพจากไฟล์ THAI.CHR มาเก็บไว้ในตัวแปร THAI ตั้งแต่ตัวแปรที่ 0-68 (THAI[0]-THAI[68])

สามารถนำมาแสดงผลภาษาไทยบนจอภาพได้ และเมื่อใช้คู่กับภาษาอังกฤษ (ไฟล์LITT.CHR) ก็ทำให้แสดงผลภาษาไทยและภาษาอังกฤษได้อย่างสมบูรณ์

โดยแนวคิดเดียวกันนี้ เราสามารถสร้างภาษาไทยในรูปแบบอื่นๆ ได้

รูปที่ 6
ภาษาไทยและภาษาอังกฤษ
ที่ได้จากโปรแกรมในรูปที่ 5

content[k] > #223 คือ วรรณยุกต์ทั้งหมด ตรวจสอบว่าถ้าข้างหน้าไม่ใช่สระบน ก็ให้กำหนดค่า yy ใหม่ เพื่อให้ภาพมาอยู่ในระดับเดียวกับสระบนนั่นเอง

การกำหนดรูปแบบของ putimage เป็น putimage(x,y+yy,Thai[asc],1); เพราะต้องการให้ภาพที่วาดไว้เดิมยังปรากฏอยู่ ซึ่งจะทำให้ข้อความเป็นไปอย่างสมบูรณ์

การนำข้อความไปแสดงก็ไม่มีความยุ่งยากดังปรากฏในโปรแกรมแล้ว

การใช้ภาษาอังกฤษร่วมด้วยนั้น ผู้เขียนอาศัยไฟล์ชื่อ LITT.CHR ที่มีอยู่แล้วโดยกำหนดขนาดเป็น 7 จากคำสั่ง Textshow(2,7,'h'); ก็สามารถแสดงภาษาอังกฤษร่วมกับภาษาไทยได้โดยใช้คำสั่ง OutTextXY(x,y,content[k]); นั่นเอง

รูปที่ 6 เป็นภาพของข้อความที่ได้จากการรันโปรแกรมในรูปที่ 5

เขียน CAI ฟิสิกส์กันดีกว่า

จากแนวคิดที่ผ่านมาเราย่อมสามารถปรับปรุงโปรแกรมในรูปที่ 1 ให้ใช้วาดภาพประกอบในการเขียน CAI ของเราได้เกือบทุกวิชา โดยถ้าต้องการภาพเต็มหน้าจอก็คงต้องกำหนดเคอร์เซอร์เป็นเพียงจุดหนึ่งจุดเท่านั้น พร้อมกับเพิ่มคำสั่งวาดภาพอื่น ๆ เข้ามาด้วย เช่น

กดแป้น S ให้จำจุด Xold,Yold ไว้

กดแป้น L ให้ลากเส้นจากจุด Xold,Yold ถึง XDraw, YDraw

กดแป้น C ให้วาดวงกลมโดยมีรัศมีเท่ากับการขจัดระหว่างจุด Xold, Yold และ XDraw, YDraw

กดแป้น B ให้วาดรูปสี่เหลี่ยม เป็นต้น

ซึ่งทำให้ได้โปรแกรมที่สมบูรณ์ขึ้นและช่วยในการวาดภาพตามที่ต้องการได้

จากฟิสิกส์ ม.4 บทที่ 3 เรื่อง สมดุล ที่กล่าวถึงการสมดุลเนื่องจากแรง 2 แรง แรง 3 แรง ฯลฯ นั้น ตามตัวอย่างที่ 3.1 หน้า 65 ได้มีโจทย์ดังนี้

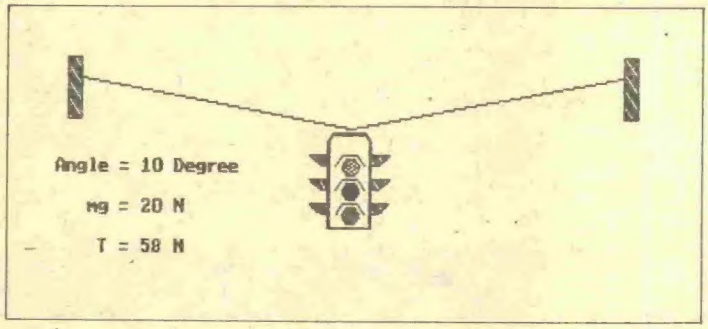
"โคมไฟจราจรแขวนไว้กลางถนนด้วยลวดสองเส้น ถ้าน้ำหนักของโคมไฟเป็น 20 นิวตัน และลวดแต่ละเส้นทำมุม 10 องศากับแนวระดับ จงหาแรงดึงในเส้นลวดทั้งสอง"

เราสามารถจำลองเหตุการณ์ในการนำโคมไฟไปแขวนครั้งนี้ให้ปรากฏในจอภาพด้วยเครื่องคอมพิวเตอร์ได้ โดยสามารถเริ่มต้นตั้งแต่ตอนที่นำโคมไฟจราจรไปแขวนกับลวดที่ถูกผูกตรึงไว้ในแนวเส้นตรง ด้วยแรงจากน้ำหนักของโคมไฟจะทำให้ลวดที่ผูกนั้นเกิดความยืดหยุ่นปรับตนเองเพื่อให้เกิดการสมดุลของแรง จึงมีผลทำให้โคมไฟจราจรนั้นเคลื่อนตัวลงจากแนวเดิม จนกระทั่งเส้นลวดทำมุม 10 องศาตามโจทย์บอก รายละเอียดของโปรแกรมดังรูปที่ 7

จากโปรแกรมในรูปที่ 7 การสร้างภาพโคมไฟจราจรเก็บไว้ในไฟล์ชื่อ Traffic.pic และเรียกภาพมาเก็บไว้ในชื่อ TrafficLamp เพื่อใช้ในแสดงการเคลื่อนที่ตอนที่แขวนโคมไฟจราจรเข้ากับเส้นลวด การคำนวณเพื่อหาระยะที่โคมไฟหย่อนลงจากแนวเดิม(h) หาได้จากมุมสุดท้ายที่โคมไฟจราจรอยู่ในสมดุล(ในโปรแกรมคือ 10 องศา)

```
                {---- PROGRAM 5.4 Force -----}
  {$I INITGR.PAS}
  {$I PICSHOW.PAS}
var TrafficLamp:pointer;
    size:word;
PROCEDURE Force;
var yy,h,mg,T:integer;
    xx,Angle:real; st:string;
begin
  xx:=200; yy:=50;   mg:=20;  Angle:=10;
  str(round(Angle),st); outtextxy(0,90,'Angle = '+st+' Degree');
  str(mg,st); outtextxy(24,110,'mg = '+st+' N');
  Angle:=Angle*pi/180;
  line(162,50,562,50);        setfillstyle(6,1);
  bar(151,40,161,70); Rectangle(151,40,161,70);
  bar(563,40,573,70); Rectangle(563,40,573,70);
  setfillstyle(1,1);  h:=round((sin(Angle)/Cos(Angle))*xx*0.7);
   repeat
    setcolor(0); line(162,50,360,yy);
                 line(562,50,360,yy);
    yy:=yy+1;
    Putimage(330,yy,TrafficLamp^,0);
    setcolor(1); line(162,50,360,yy);
                 line(562,50,360,yy);
   until yy>=50+h;
    T:=round(mg/(2*sin(Angle))); str(T,st);
    outtextxy(32,130,'T = '+st+' N');
end; {Force}

{==================== Main ======================}
 begin
  opengraph;
   size:=imagesize(0,0,75,50); getmem(TrafficLamp,size);
  OpenFile('Traffic.pic');
    ShowPic(330,50,0);
    getimage(330,50,405,100,TrafficLamp^);
  close(PicFile);
    Force;
  repeat until keypressed;
  closegraph;
 end.
```

รูปที่ 7
โปรแกรมเรื่อง
การสมดุลของแรง



รูปที่ 8 ภาพที่ได้จากโปรแกรมในรูปที่ 7

คำนวณหาค่าของแรงดึงในเส้นลวด โดยการแตกแรง T ออกไปเป็นแรงย่อย และได้แรงที่อยู่ในแนวดิ่งรวมกันมีค่าเป็นศูนย์ หรือ mg = 2TSIN(Angle) โดยผู้เขียนทิ้งส่วนนี้ไว้ให้ผู้อ่านลองเขียนโปรแกรมเพิ่มเติมเอง

เมื่อโปรแกรมทำงานสิ้นสุดก็จะมีข้อความบอกให้ทราบถึงค่าของแรงดึงในเส้นลวด(ดังรูปที่ 8) ผู้อ่านอาจลองเปลี่ยนค่าของตัวแปร mg, Angle ให้เป็นค่าอื่น ๆ แล้วรันดู ก็จะได้ค่าตามต้องการ สิ่งที่ควรทำอีกอย่างหนึ่งก็คือค่าของ T ที่คำนวณได้นั้นโปรแกรมกำหนดไว้เป็นตัวแปรชนิด integer จึงทำให้ได้ค่า T เป็นเลขจำนวนเต็มเท่านั้น ให้ลองแก้ไขเป็นเลขจำนวนจริง(real)ซึ่งจะให้ค่าของ T ใกล้เคียงกับความเป็นจริงมากขึ้น

ไฟล์ PICSHOW.PAS เป็นไฟล์ที่คล้ายกับโปรแกรมในรูปที่ 3 กล่าวคือ มีชุดคำสั่งอยู่ 2 ชุดคำสั่งคือ OpenFile และ ShowPic เพียงแต่ดัดแปลงชุดคำสั่ง ShowPic โดยตัดค่า Mode ออกไปจึงเหลือเพียง ShowPic(x,y:integer; PicNo: byte); และกำหนดเลข 1 แทนตัวแปร Mode ในการ PutImage ซึ่งทำให้คำสั่งเป็น PutImage(x,y,Picbuff,1); เลย เมื่อตัดส่วนตั้งแต่ Main จนจบโปรแกรมออกไปแล้วเก็บไว้ในชื่อ PICSHOW.PAS นำมาใช้งานได้โดยการใช้คำสั่ง {$I PICSHOW.PAS} ดังที่เห็นในโปรแกรมในรูปที่ 7

ถึงขั้นนี้แล้วท่านมีแนวคิดจะเขียน CAI วิชาฟิสิกส์บ้างหรือยัง

แม้ว่าตอนนี้จะระบุว่าเป็นตอนจบ แต่การเขียนโปรแกรมด้าน CAI นั้นจะยังคงอยู่ไม่มีวันสุดสิ้นดังนั้นงานการพัฒนาโปรแกรมการเรียนการสอนยังสามารถกระทำได้เรื่อย ๆ อย่างไม่มีวันที่สิ้นสุด ซึ่งอยู่ที่มุมมองของคนเขียนโปรแกรมคอมพิวเตอร์ว่าจะเห็นแนวทางพัฒนาเอง

พบกันใหม่ในโปรแกรมการเรียนการสอนวิชาฟิสิกส์ตามปกติในฉบับต่อ ๆ ไป